

Tutorial on AI Planning for Robotics: Beyond Blocksworld

06.02.2025
Shin Watanabe

Acknowledgement

- Content inspired by material and discussions with people organizing and attending the [ICAPS 2024 summer school](#) (International Conference on Planning and Scheduling)
- [Tutorial on Task and Motion Planning](#) by Guy Azran
- [PDDL Playground App](#) by me & Victor Paléologue (at Softbank Robotics)



Plan

- **Why** should I care?

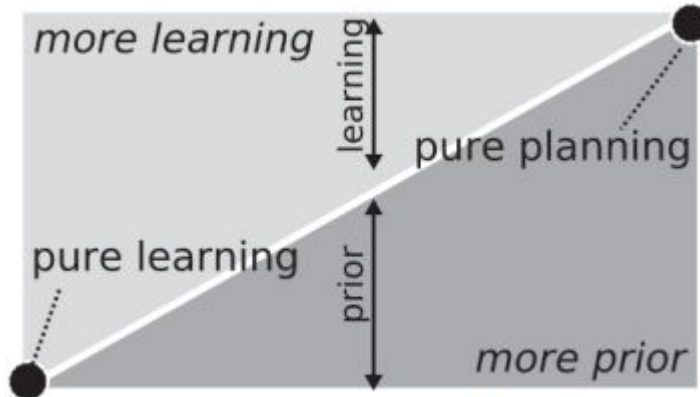
- **What** is the big deal?

- **How** does it work?

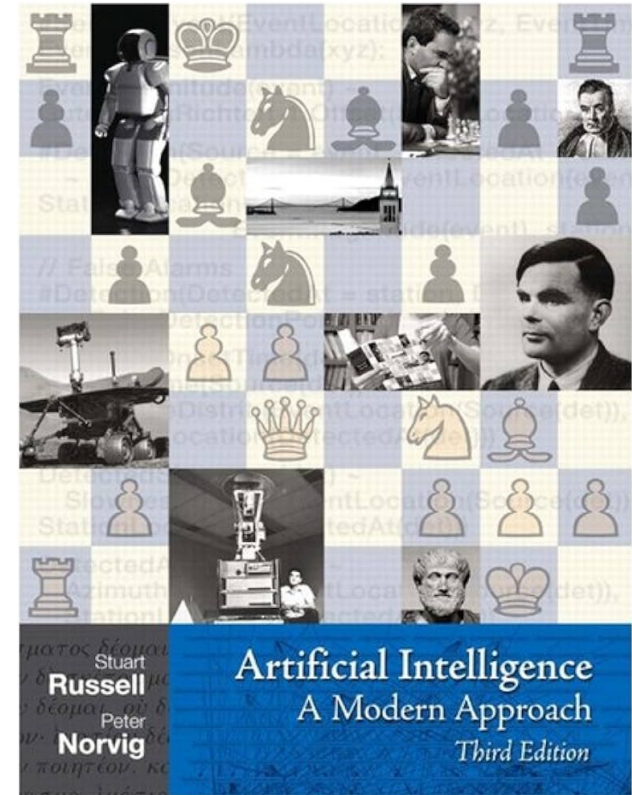
What is AI planning?

a.k.a. automated planning
a.k.a. good old-fashioned AI (GOFAI)
a.k.a. “It’s just if-statements”

“Planning is the art and practice of thinking before acting”
- Patrik Haslum



Source: Oliver Brock



The book features the same amount of content for ML as Knowledge, Representation and Reasoning

Why should we use planning for robotics?

“I thought it failed in the 70s”

Rephrased question:

“Do we want our robots to think before they act?”
(thinking = imagining and evaluating possible future scenarios)

My short answer: yes, when it encounters situations it has never seen

My long answer:

- It should learn to recognize *when* it must think
 - It should think *in order to* speed up learning
 - It should learn *in order to* speed up thinking
- ⇒ integrated planning and learning



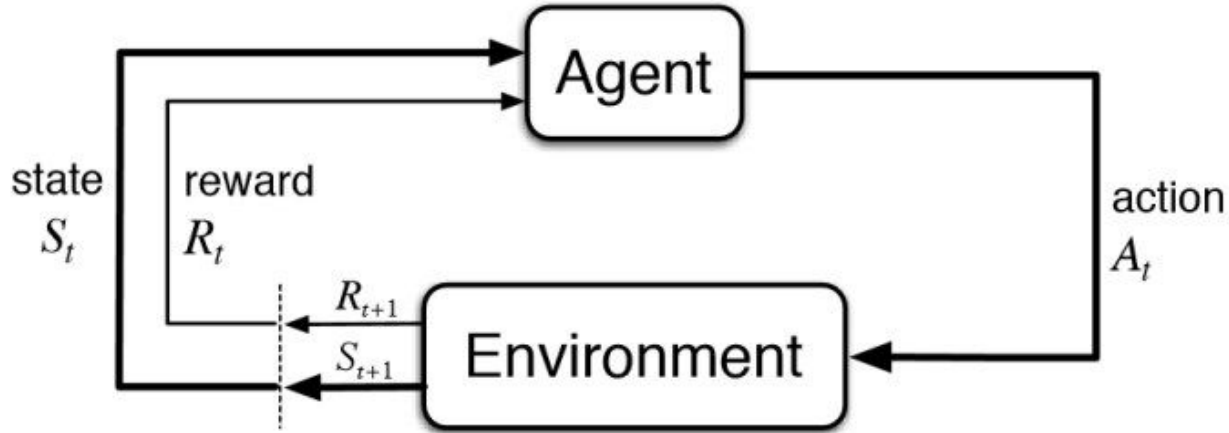
Shakey the robot

Why should we use planning for robotics?

Provocative Answer:

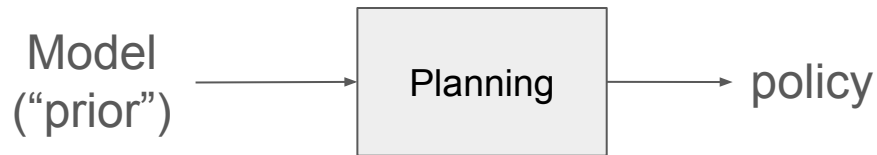
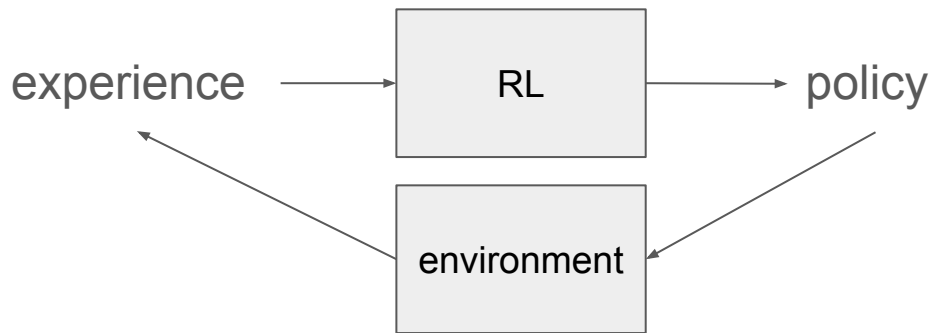
Many ML researchers are *already* using planning
without realizing it

Casting simulation-based RL as planning

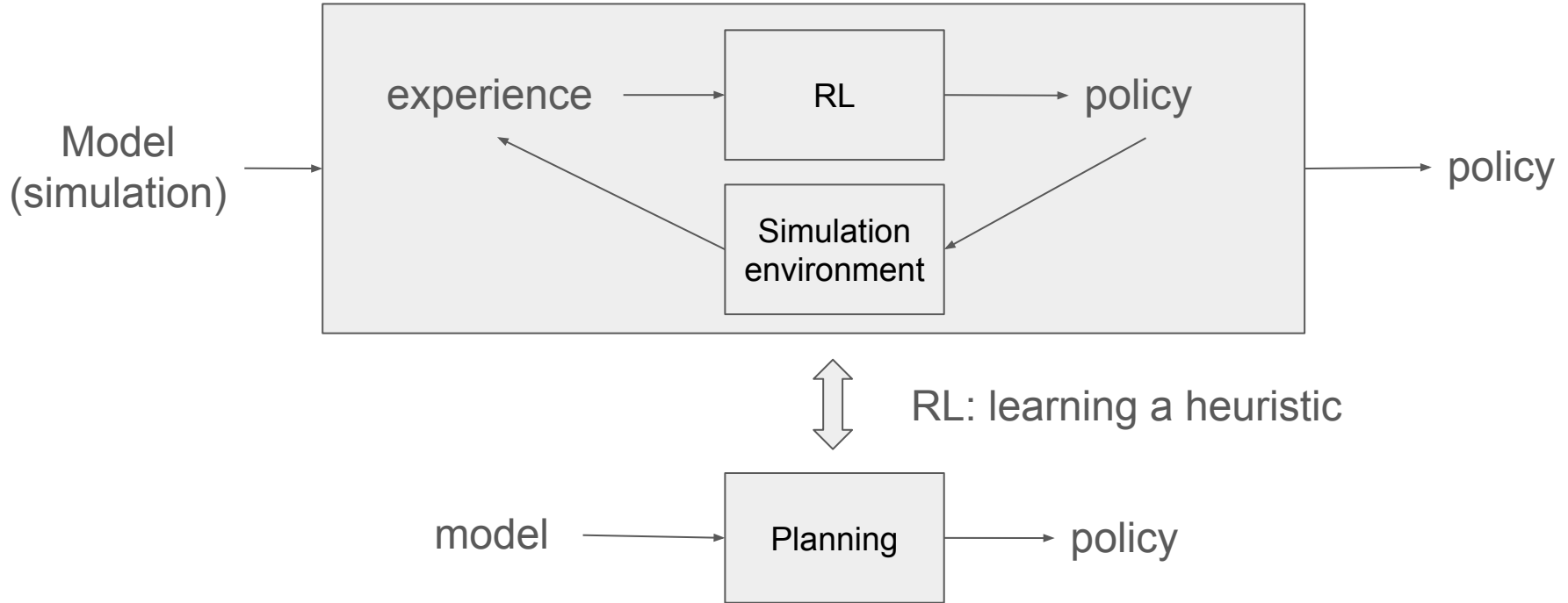


Sutton and Barto, 1998

Casting simulation-based RL as planning



Casting simulation-based RL as planning



Casting simulation-based RL as planning

Implications:

- Planning researchers can use many RL/ML tools for their own work
 - Large research subfield focused on learning good heuristics for search
- RL researchers can use many planning tools for their own work
 - Some notable examples (alphaGo line), but few (that I found) going further than search

Mastering the game of Go with deep neural networks and tree search

[David Silver](#) , [Aja Huang](#), [Chris J. Maddison](#), [Arthur Guez](#), [Laurent Sifre](#), [George van den Driessche](#), [Julian Schrittwieser](#), [Ioannis Antonoglou](#), [Veda Panneershelvam](#), [Marc Lanctot](#), [Sander Dieleman](#), [Dominik Grewe](#), [John Nham](#), [Nal Kalchbrenner](#), [Ilya Sutskever](#), [Timothy Lillicrap](#), [Madeleine Leach](#), [Koray Kavukcuoglu](#), [Thore Graepel](#) & [Demis Hassabis](#) 

[Nature](#) **529**, 484–489 (2016) | [Cite this article](#)

Search on the Replay Buffer: Bridging Planning and Reinforcement Learning

Benjamin Eysenbach ^{$\theta\phi$} , Ruslan Salakhutdinov ^{θ} , Sergey Levine ^{$\phi\psi$}
 ^{θ} CMU, ^{ϕ} Google Brain, ^{ψ} UC Berkeley
beysenba@cs.cmu.edu

AI Planning isn't just about *search*!

Plan

- Why should I care?

- What is the big deal?

- How does it work?

What is so great about AI planning in robotics?

- It can deal with **relational** representations (predicate logic) *
 - * There exist recent ML-based approaches that can deal with relational representations
- Planners can solve tasks requiring **many objects** and **long skill sequences** *
 - * Usually given a **fully observable** environment and **deterministic** dynamics

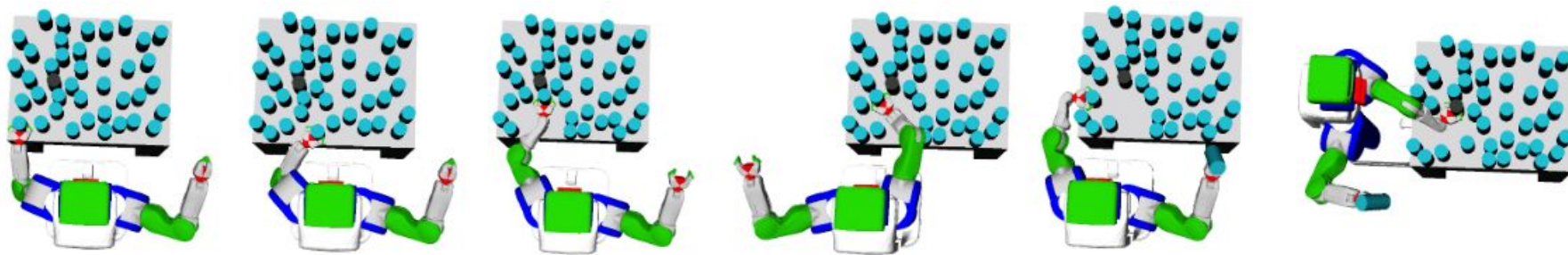


Fig. 6: Some of the grasps executed while solving an instance of the 40 object cluttered table with the dark object as the target.

Plan

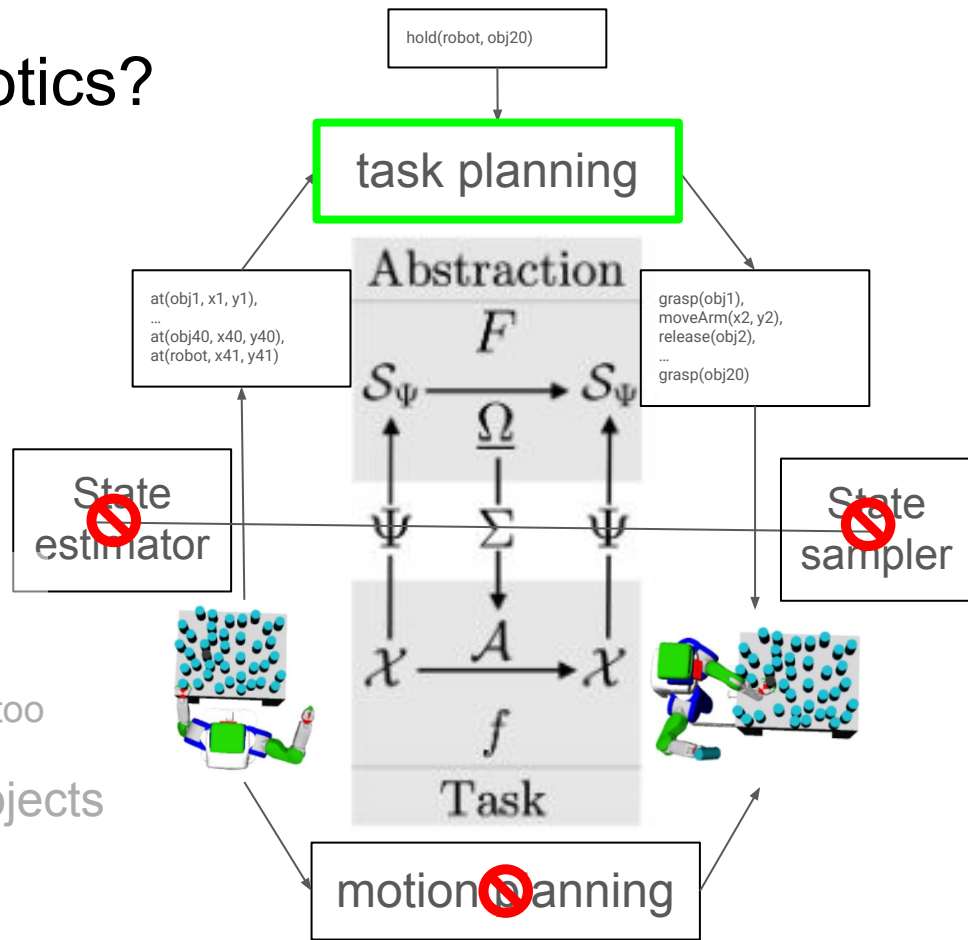
- Why should I care?

- What is the big deal?

- How does it work?

How is planning used in robotics?

- Robotics entails two types of planning:
task planning and *motion planning*
 - *Task-and-motion planning, bilevel planning*
- The environment must be perceived and bound to symbols
 - Learning-based approaches to the rescue!
- The symbolic actions must be bound to motions
 - Learning-based approaches can help here too
- How to represent relations between objects and actions symbolically?
 - e.g. What can you do with a cylinder?



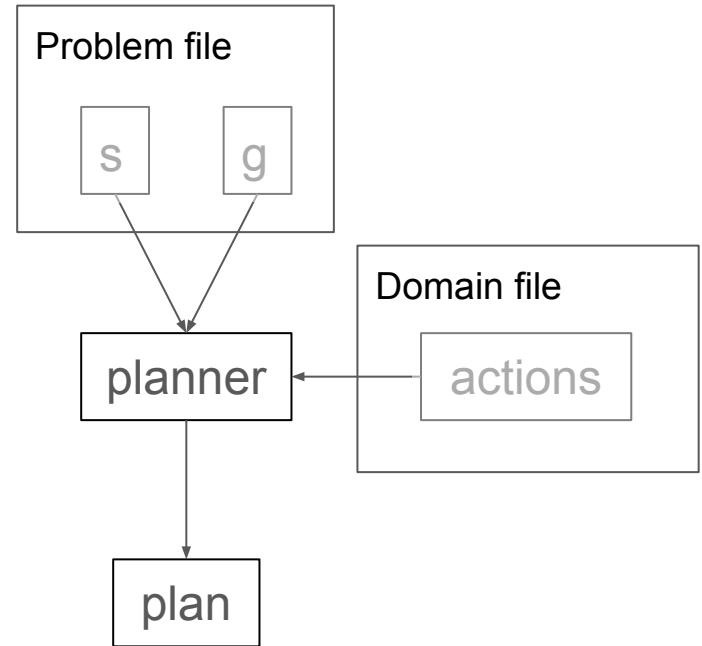
Domain Independent Planning

Core philosophy: the planner should know *nothing* about the problem they are solving \longleftrightarrow Opposite of ML!

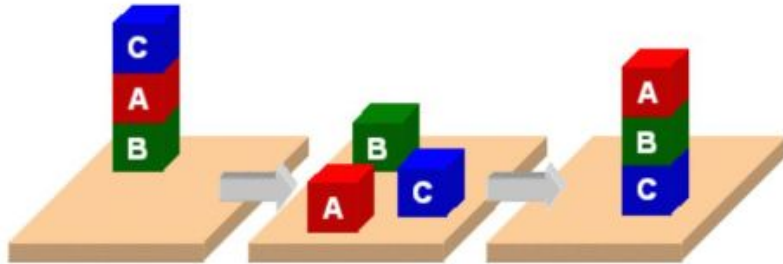
All we need is:

1. the starting state
2. the actions we can apply
3. the desired end state

These are written in the Planning Domain Definition Language (PDDL), then parsed into the planner



Let's write a planning domain / problem!



Blocksworld domain [UMBC]

Instances correct	
GPT-4	I-GPT3
206/600 (34.3%)	41/600 (6.8%)

LLM performance on blocksworld

Let's run it in a physics engine!

Blocksworld with multiple tables!

Running on Mujoco via PDDLGym

[Link to sample video](#)

Useful links & resources

Web-based PDDL repository, editor & solver: <https://editor.planning.domains/#>

PDDL wiki: <https://planning.wiki>

Python package with PDDL tools: <https://unified-planning.readthedocs.io/>
([AIPlan4EU](#))

Intro to AI planning (course from KCL Inf. Dept.):
<https://www.youtube.com/watch?v=EeQcCs9SnhU>